

# MEMORIA ESCRITA DEL PROYECTO

CFGS Desarrollo de Aplicaciones Web

InternJobs

**Autor:** Óscar Serrano Molina

**Convocatoria:** 2º Semestre curso 24/25

## Índice de contenidos

<b>1. INTRODUCCIÓN</b>	<b>3</b>
1.1. Motivación	3
1.2. Abstract	4
1.3. Objetivos propuestos (generales y específicos)	4
<b>2. Estado del Arte</b>	<b>5</b>
<b>3. METODOLOGÍA USADA</b>	<b>6</b>
<b>4. TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS EN EL PROYECTO</b>	<b>7</b>
<b>5. PLANIFICACIÓN, DIAGNÓSTICO Y CONTEXTO LABORAL</b>	<b>8</b>
<b>6. ANÁLISIS DEL PROYECTO</b>	<b>11</b>
<b>7. DISEÑO DEL PROYECTO</b>	<b>17</b>
<b>8. DESPLIEGUE Y PRUEBAS</b>	<b>23</b>
<b>9. CONCLUSIONES</b>	<b>27</b>
<b>10. VÍAS FUTURAS</b>	<b>29</b>
<b>11. BIBLIOGRAFÍA/WEBGRAFÍA</b>	<b>31</b>
<b>12. ANEXOS</b>	<b>32</b>
12.1. Manual de Usuario	32
12.1. Manual de Instalación	38

## 1. Introducción

### 1.1. Motivación

En el transcurso de los últimos cursos académicos he observado una problemática creciente en torno al acceso de los estudiantes de ciclos formativos a las prácticas laborales. Aunque los centros de formación son los encargados de cubrir esta demanda y facilitar el proceso, la realidad es que cada vez resulta más habitual que no se logre cubrir la demanda de prácticas al 100% del alumnado. Esta situación genera incertidumbre y obstáculos en un momento clave de transición hacia el mercado laboral.

Por otro lado, también se está produciendo un cambio en las expectativas del propio alumnado. Los alumnos ya no buscan únicamente cumplir con un trámite formativo, sino que desean encontrar unas prácticas laborales que estén alineadas con sus intereses y con su proyección profesional. Esto hace que muchos opten por buscar oportunidades por su cuenta, lo cual supone un desafío adicional dada la escasa existencia de herramientas específicas para ello.

A nivel tecnológico he notado un vacío claro: en España, apenas existen soluciones web diseñadas específicamente para conectar estudiantes con empresas con el objetivo final de formalizar un **convenio de colaboración**. En su lugar se están utilizando portales de empleo generalistas que, si bien son útiles para el mercado laboral ordinario, apenas dedican espacio a las ofertas de prácticas formativas. El número de ofertas específicas para estudiantes en estos portales es muy limitado y se suele encontrar mal categorizado. A menudo se suelen confundir las ofertas específicas para estudiantes de ciclos formativos con las ofertas para becarios, estudiantes de grados universitarios, así como con las ofertas orientadas a juniors.

Además, desde la perspectiva de las empresas, especialmente aquellas que dependen habitualmente de perfiles en prácticas, tampoco existe un espacio claro donde publicar este tipo de ofertas sin que acaben mezcladas y confundidas con ofertas del mercado laboral ordinario. Esta falta de segmentación dificulta tanto la publicación como la recepción de candidaturas adecuadas.

Ante este contexto surge **InternJobs** como una propuesta que busca cerrar la brecha entre estudiantes, centros de formación y empresas. Una plataforma pensada específicamente para conectar candidatos en búsqueda de prácticas con empresas que están dispuestas a ofrecerlas, bajo un enfoque claro y segmentado, adaptado a las necesidades reales de ambas partes.

## 1.2. Abstract

In recent years, many educational institutions have struggled to ensure internship placements for all their students, despite being responsible for coordinating these opportunities. At the same time, students are increasingly seeking internships aligned with their personal career goals, a match that is not always guaranteed through school-provided companies.

Additionally, there is a lack of dedicated digital platforms in Spain focused exclusively on internship opportunities. As a result, both students and companies must rely on general job portals, which offer only a limited number of relevant internships listings. Companies aiming to recruit interns through training agreements (“convenios de colaboración” in Spain) also lack a clear space to post these offers, often mixing them with standard job listings aimed at junior professionals.

**InternJobs** was created to address these challenges by offering a specialized web platform that connects students with companies through collaborative internship agreements.

## 1.3. Objetivos

Antes de comenzar el desarrollo de **InternJobs**, se establecieron una serie de objetivos que permitieran dar solución a la problemática que detecté en el proceso de encontrar y formalizar las prácticas en empresa por parte de los estudiantes de ciclos formativos. A continuación se presentan los objetivos generales y específicos que han guiado el diseño, implementación y despliegue de la plataforma.

### Objetivos generales:

- Diseñar y desarrollar una plataforma web que facilite el contacto entre estudiantes y empresas para la gestión de prácticas profesionales mediante convenios de colaboración.
- Ofrecer una solución intuitiva, accesible y funcional tanto para los candidatos como para las empresas, optimizando el proceso de publicación, búsqueda y gestión de ofertas.
- Proporcionar una herramienta digital específica para el ámbito de prácticas laborales, cubriendo el vacío existente en el mercado actual, especialmente en el contexto educativo español.

Objetivos específicos:

- Desarrollar un sistema de registro y autenticación para dos tipos de usuarios: Candidatos y Empresas.
- Implementar formularios que permitan a los candidatos gestionar su perfil personal y académico, así como postularse a las ofertas disponibles.
- Permitir a empresas publicar ofertas y gestionar las candidaturas recibidas.
- Diseñar un panel privado diferenciado para candidatos y empresas con funciones adaptadas a cada tipo de usuario.
- Crear un sistema de filtrado y búsqueda de ofertas de prácticas laborales accesible y funcional.
- Establecer una arquitectura backend segura que gestione correctamente la persistencia de datos y la autenticación de usuarios.
- Ofrecer una interfaz con una experiencia de usuario moderna y fluida.

## 2. Estado del Arte

La búsqueda de empleo experimentó una profunda transformación con la llegada de internet. A mediados de los 90 surgieron las primeras bolsas de empleo online, que permitían a los candidatos acceder a ofertas laborales desde sus hogares y enviar sus currículums de forma digital. Portales como **Monster** (1994) en EEUU, o más tarde **InfoJobs** (1998) en España, marcaron un antes y un después en el acceso al mercado laboral, eliminando las barreras geográficas y facilitando una conexión directa entre empleadores y candidatos.

Durante la década de los 2000 y 2010, esta evolución continuó con la aparición de plataformas más especializadas y dinámicas, como **LinkedIn** (2002), que no solo ofrecía ofertas laborales, sino que también permitía construir una red profesional y visibilizar el perfil de candidatos y empresas de forma más activa y estratégica, y con un alcance de nivel internacional. A su vez, surgieron plataformas enfocadas en sectores concretos, como **Domestika** para empleos creativos o **Tecnoempleo** para perfiles IT.

Sin embargo, a pesar de esta evolución, las plataformas dedicadas exclusivamente a prácticas profesionales o colaboración educativa son escasas, especialmente en el mercado español. Los portales de empleo generalista suelen incluir algunas ofertas de prácticas, pero de forma escasa, a menudo sin filtros específicos ni funcionalidades adaptadas a las necesidades de estudiantes y empresas. Esto obliga a los alumnos a buscar oportunidades en entornos poco optimizados para esta situación académica y profesional.

Actualmente existen algunas soluciones como **Talentoteca**, promovida por fundaciones universitarias, o los portales de prácticas externas de universidades concretas, pero su alcance es limitado y no están abiertas al público general. Además, se orientan principalmente a estudiantes universitarios, dejando fuera al gran número de alumnos de formación profesional que también requieren encontrar prácticas en empresa para completar su formación.

**InternJobs** surge precisamente para cubrir ese vacío: una plataforma especializada en prácticas profesionales bajo convenio de colaboración, dirigida específicamente a los estudiantes de ciclos formativos de grado medio y grado superior. Esta distinción permite a InternJobs adaptar su lenguaje, estructura y funcionalidad a las necesidades reales de este colectivo, tanto desde el punto de vista del alumno como el de las empresas que requieren incorporar este tipo de talento. Su orientación específica y facilidad de uso la posicionan como una solución diferenciadora dentro del panorama actual.

### 3. Metodología del desarrollo

Para el desarrollo de **InternJobs** he optado por seguir el modelo de desarrollo en cascada, una metodología tradicional caracterizada por su estructura secuencial, donde cada fase del proyecto debe completarse antes de iniciar la siguiente. Esta elección se justifica por la naturaleza del proyecto, que desde su concepción contó con requisitos claros, un alcance bien definido y una idea precisa del flujo de la aplicación.

Dado que el desarrollo ha sido realizado por un único desarrollador, el modelo en cascada ofrecía una ventaja significativa en cuanto a la organización y documentación, permitiendo mantener una visión clara del progreso y de las tareas pendientes. Aunque en la práctica se realizaron iteraciones y ajustes propios de metodologías más flexibles, la base estructurada del modelo en cascada sirvió como guía para avanzar con orden y coherencia.

Las fases seguidas fueron:

- **Requisitos y análisis:** Se definieron las necesidades tanto de candidatos como de empresas estableciendo las funcionalidades que ofrecería la plataforma.
- **Diseño del sistema:** Se planificó la arquitectura general de la aplicación, dividiendo claramente el frontend del backend con una arquitectura MVC y una interfaz SPA. Se detallaron las tecnologías a utilizar.

- **Implementación:** Se desarrollaron las distintas funcionalidades, priorizando aquellas que permitieran validar el flujo completo de la plataforma desde el inicio. En esta parte se diferenció el desarrollo del backend y del frontend.
- **Pruebas:** Se realizaron las pruebas funcionales de cada módulo para todos los casos de uso, verificando la lógica de negocio y autenticación, así como las posibles interacciones entre usuarios y datos. En la parte del backend se comprobaron todos los endpoints con POSTMAN.
- **Mantenimiento:** Una vez se completa la primera versión funcional, se procedió a detectar errores, aplicar mejoras, ajustar comportamientos y preparar el proyecto para su entrega académica, incluyendo pequeñas refactorizaciones y revisiones finales.

## 4. Tecnologías y herramientas utilizadas

### Frontend:

- [Vue.js 3](#): Framework progresivo de JavaScript utilizado para construir la interfaz de usuario de la aplicación. Su enfoque declarativo y basado en componentes facilita la escalabilidad y mantenimiento del proyecto.
- **Vue Router**: Librería oficial de enrutamiento para Vue 3 , que permite la navegación entre distintas vistas de la aplicación de manera fluida y sin latencia.
- **Pinia**: Sistema de gestión de estado oficial para Vue 3, utilizado para centralizar y sincronizar los datos compartidos entre componentes.
- **TailwindCSS**: Framework de estilos basado en utilidades que ha permitido estilar la interfaz de forma moderna, responsiva y coherente, especialmente adaptada al estilo oscuro que utiliza.

### Backend:

- [Node.js](#): Entorno de ejecución para JavaScript del lado del servidor, elegido por su rendimiento, facilidad de uso y integración natural con Express.

- **Express.js**: Framework minimalista de [Node.js](#) que ha facilitado la creación de la API RESTful que compone la arquitectura del backend de la aplicación.
- **PostgreSQL**: Base de datos utilizada para la persistencia de datos de manera estructurada y relacional. Con gran soporte, escalabilidad y posibilidades de hosting.
- **TablePlus**: Sistema gestor de base de datos, intuitivo y minimalista con gran soporte y comunidad.
- **Postman**: Herramienta empleada para probar manualmente los endpoints del backend durante el desarrollo y validación de la API.

#### **Seguridad:**

- **JWT**: Mecanismo de autenticación y autorización mediante tokens, clave en la comunicación frontend-backend, utilizado para proteger rutas privadas y sistema de login tanto de candidatos como de empresas.
- **Bcrypt**: Sistema de encriptación utilizado para proteger las contraseñas mediante hashing antes de ser almacenadas en la base de datos.

#### **Otras herramientas:**

- **Git y GitHub**: Control de versiones del código y alojamiento del repositorio en la nube para mantener un historial de desarrollo limpio y poder compartir el código fácilmente. Me ha permitido trabajar desde diferentes PCs y lugares.

## 5. Planificación, diagnóstico y contexto laboral

La planificación se ha llevado a cabo mediante la metodología en cascada, lo cual ha hecho que se organice el desarrollo de manera modularizada y secuencial, de manera que el fin de la primera etapa daba paso a la segunda y así sucesivamente.

El proyecto comienza de facto el día 04 de Marzo de 2025, y obtenemos la primera versión entregable el día 01 de Junio de 2025, y continúa de manera iterativa, tal y como ofrece el modelo en cascada, el mantenimiento y las mejoras. Dado que la complejidad de la aplicación se encuentra en el código, y ha sido la parte del desarrollo que mayor tiempo ha conllevado.



En la siguiente tabla se realiza el desglose de horas invertidas en cada partes el proyecto:

Fase del proyecto	Fecha de inicio	Horas totales
Toma de requisitos	04/03/2025	20
Diseño de la aplicación	18/03/2025	45
Implementación (Backend)	06/04/2025	55
Implementación (Frontend)	26/04/2025	70
Pruebas	18/05/2025	35
Plan de Mantenimiento	30/05/2025	5
	<b>Total horas:</b>	<b>230</b>

Figura 1 - Tabla de estimación de horas de desarrollo

Para tener un monitoreo exhaustivo del desarrollo de la aplicación se ha utilizado el diagrama de Gantt, permitiendo de una manera visual estimar el tiempo que ha llevado cada etapa del desarrollo en forma de días, y aunque no todos los días se ha podido invertir la misma cantidad de horas en el proyecto, la estimación visual es un acercamiento bastante cercano a la realidad:

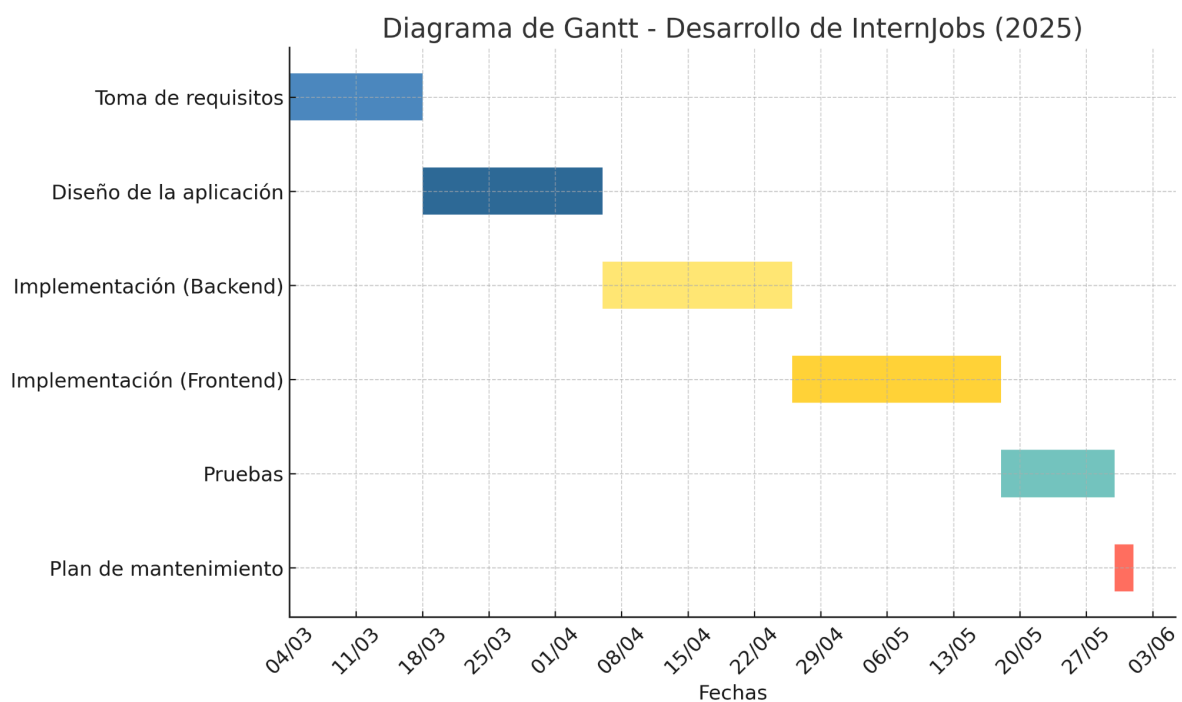


Figura 2 - Diagrama de Gantt para el desarrollo de InternJobs

Tanto en la tabla del desglose de horas como en el diagrama de Gantt, no se tienen en cuenta la cantidad de horas invertidas en la formación, que he debido recibir para prepararme de manera adecuada, en las tecnologías que utiliza el proyecto, ya que algunas no forman parte de la titulación cursada (DAW), todo y tratarse de tecnologías muy asentadas en el mercado, y que debemos estimar en unas 200 horas, tanto en el periodo previo al inicio del proyecto (19/02/2025 - 04/03/2025) como durante el desarrollo del proyecto.

En el caso del diagnóstico del proyecto he decidido basarme en el análisis **DAFO**, debido a que se trata de una herramienta estratégica y esquemática que nos ayuda de forma rápida a obtener una estimación de la viabilidad de mercado del proyecto.

Debilidades	Fortalezas
Competencia con otros portales de empleo consolidados.	Plataforma especializada en prácticas profesionales.
Dependencia de un número considerable de usuarios activos para generar valor.	Conexión directa entre empresas y estudiantes para formalizar convenios de colaboración.
Recursos limitados para marketing y visibilidad inicial.	Diseño minimalista e intuitivo orientado a usuarios jóvenes.
Oportunidades	Amenazas
Aumento de la demanda de prácticas curriculares adaptadas a la carrera del estudiante.	Entrada de nuevas plataformas con más presupuesto y respaldo institucional.
Potenciales alianzas entre centros de formación y empresas.	Cambios en la normativa educativa sobre las prácticas en empresas.
Mayor digitalización en procesos de selección y formación.	Desinterés del público objetivo por saturación del mercado digital.

Figura 3 - Tabla de análisis DAFO.

El desarrollo de InternJobs se enmarca dentro de un contexto laboral caracterizado por la transformación digital del empleo juvenil, la necesidad de experiencia práctica para la inserción profesional y la creciente demanda de plataformas que faciliten estos procesos.

**Entorno educativo-empresarial:**

El proyecto parte de la necesidad real de mejorar el vínculo entre estudiantes y empresas mediante convenios de colaboración. La plataforma actúa como puente entre ambos, simplificando el proceso administrativo y visibilizando oportunidades.

**Trabajo remoto y digitalización:**

InternJobs promueve que los procesos de postulación y gestión de ofertas se realicen 100% online, adaptándose al trabajo remoto y al presente y futuro del mercado.

**Innovación y formación técnica:**

El proyecto ha supuesto un proceso continuo de aprendizaje técnico y profesional, desde el diseño de la interfaz hasta la implementación del backend con autenticación mediante JWT, integración de formularios y almacenamiento seguro de datos.

## 6. Análisis del proyecto

A continuación se van a detallar las especificaciones, funcionalidades y restricciones de la aplicación. Por lo que primero enumeraremos los requisitos funcionales y no funcionales del proyecto.

**Requisitos funcionales:**

- **Registro de usuarios:** Los usuarios pueden registrarse en la aplicación diferenciando entre dos perfiles (Candidatos y Empresas), con información específica para cada perfil.
- **Actualización de perfiles:** Los usuarios, independientemente de su perfil pueden actualizar los datos referentes a su perfil.
- **Publicación de ofertas:** Los usuarios con perfil de empresa pueden subir sus ofertas de prácticas mediante el formulario destinado a ello.
- **Actualización o eliminación de ofertas:** Los usuarios con perfil de empresa pueden acceder a sus ofertas publicadas y actualizar los datos referentes a esta, o bien, eliminar dicha oferta.

- **Aplicación a ofertas:** Los usuarios con perfil de candidato pueden acceder, y si lo requieren, aplicar a las ofertas publicadas.
- **Gestión de candidaturas:** Los usuarios con perfil de empresa pueden acceder a las candidaturas recibidas a través de sus ofertas publicadas y gestionar el estado de estas.
- **Perfil privado:** Los diferentes perfiles de la aplicación pueden acceder a su panel privado “Dashboard” desde el cual pueden consultar o gestionar su información.
- **Seguridad:** El sistema ha sido diseñado de manera que cada usuario registrado únicamente puede acceder y modificar la información propia de su perfil, con un sistema de login mediante JWT.

#### **Requisitos no funcionales:**

- **Rendimiento:** La aplicación debe funcionar sin tiempos de latencia demasiado dilatados en el tiempo, debe cargar la home page de manera rápida y la navegación entre secciones y perfil debe notarse ágil y fluida.
- **Escalabilidad:** La aplicación debe estar diseñada pensando en un aumento considerable de la cantidad de usuarios, permitiendo adaptarse al crecimiento de manera ágil.
- **Seguridad:** Los datos de los usuarios deberán estar protegidos, los password debidamente hasheados y las rutas privadas protegidas.
- **Usabilidad:** Diseño de la interfaz intuitiva y fácil, con responsividad a diferentes tamaños de pantalla.
- **Accesibilidad:** La aplicación ha sido diseñada de manera accesible para diferentes tipos y niveles de discapacidad.
- **Mantenibilidad:** La aplicación está construida por un código modularizado de manera intuitiva, estructurado y documentado, facilitando así el mantenimiento de esta.

Estos son los requisitos que han dado forma al diseño de la aplicación, guiando la funcionalidad de esta y marcando las especificaciones.

## Diseño de la base de datos y modelo E-R:

Diagrama E-R:

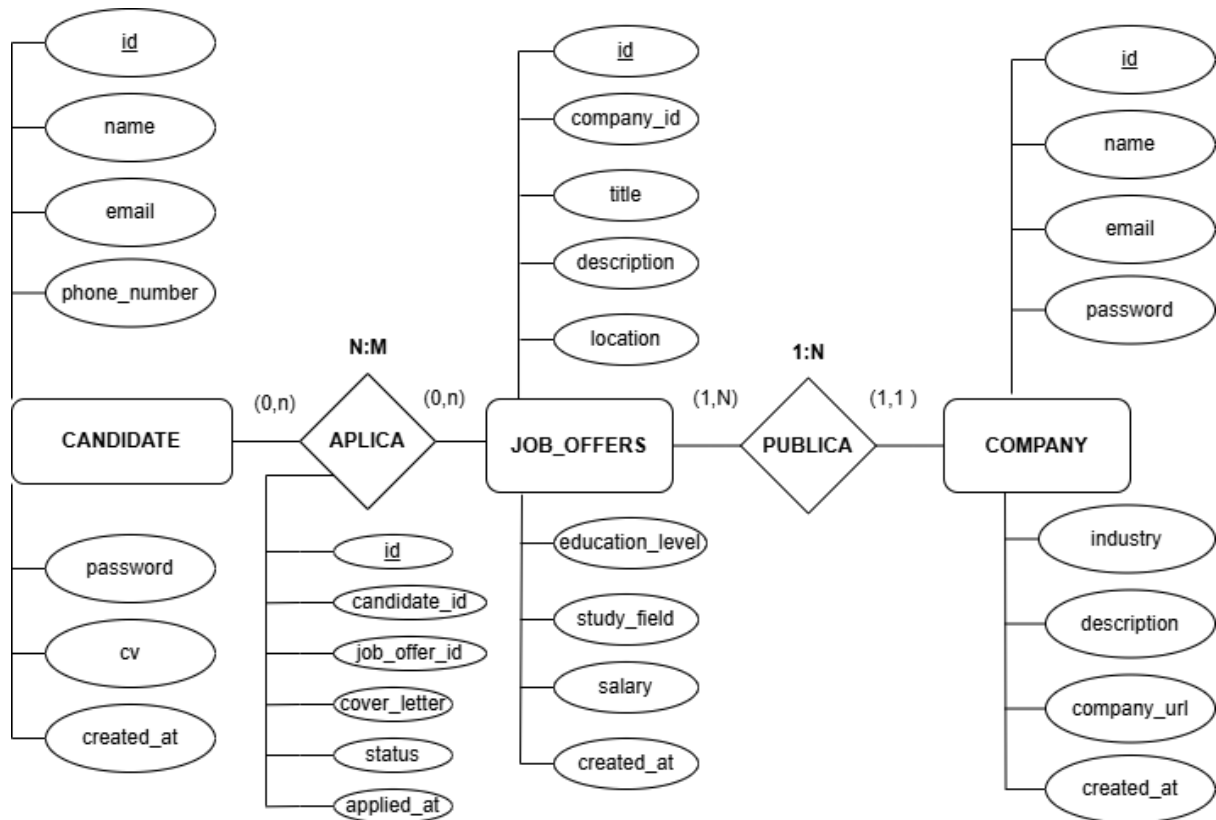


Figura 4 - Modelo E-R.

Tablas normalizadas:

- candidates:
  - id (PK)
  - name
  - email
  - phone\_nomber
  - password
  - cv
  - created\_at
- companies:
  - id (PK)
  - name
  - email
  - password
  - industry
  - description
  - company\_url
  - created\_at
- job\_offers:
  - id (PK)
  - company\_id (FK)
  - title
  - description
  - location
  - education\_level
  - study\_field
  - salary
  - created\_at
- applications:
  - id (PK)
  - candidate\_id (FK)
  - job\_offer\_id (FK)
  - cover\_letter
  - status
  - applied\_at

Como podemos observar desde el modelo E-R hasta la normalización de la base de datos obtenemos una tabla auxiliar “applications” para normalizar la relación con cardinalidad N:M de “aplica”.

Definitivamente obtenemos el modelo físico normalizado:

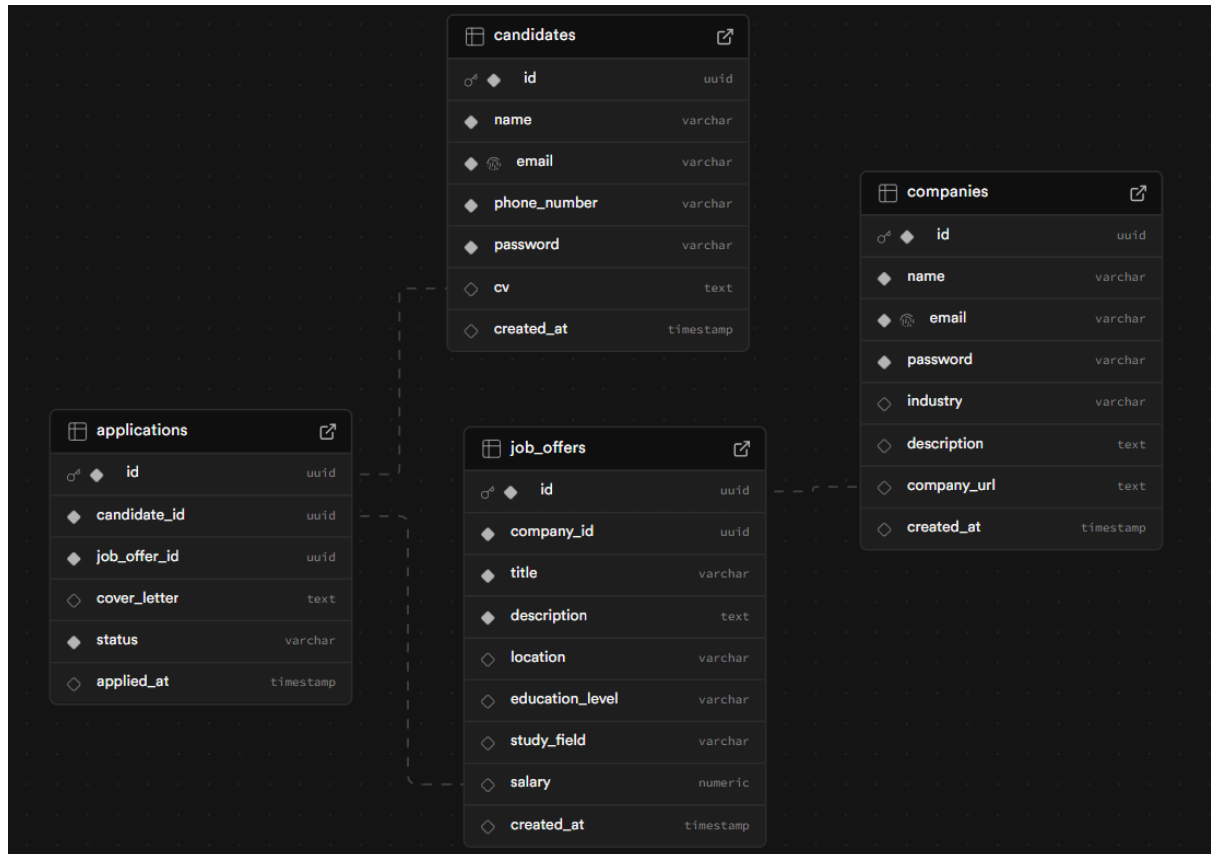


Figura 5 - Modelo físico de la base de datos.

### Actores:

- Candidato
- Empresa
- Usuario no registrado

### Casos de uso Candidato:

- Registrarse en la aplicación: Crear un nuevo perfil con toda la información necesaria para el candidato.
- Login/Logout: Acceder a la cuenta de candidato en la aplicación y salir de la cuenta.

- Acceder al perfil privado: Acceder al perfil para consultar la información relacionada al usuario.
- Actualizar perfil: Editar y guardar la información del candidato.
- Buscar y filtrar ofertas: Acceder a las ofertas publicadas y filtrarlas según varios parámetros.
- Ver detalle de las ofertas: Acceder a la información extendida de cada oferta.
- Aplicar a oferta: Postularse como candidato a una oferta.
- Incluir carta de presentación: Redactar e incluir una carta de presentación en el momento de aplicar a la oferta.
- Consultar el estado de las candidaturas: Consultar la lista de las ofertas a las que el candidato ha aplicado y el estado en el que se encuentran.

#### **Casos de uso para Empresa:**

- Registrarse en la aplicación: Crear un nuevo perfil con toda la información necesaria para la empresa.
- Login/Logout: Acceder a la cuenta de empresa en la aplicación y salir de la cuenta.
- Acceder al perfil privado: Acceder al perfil para consultar la información relacionada a la empresa.
- Actualizar perfil: Editar y guardar la información de la empresa.
- Buscar y filtrar ofertas: Acceder a las ofertas publicadas y filtrarlas según varios parámetros.
- Ver detalle de las ofertas: Acceder a la información extendida de cada oferta.
- Publicar oferta: Crear y publicar oferta de prácticas.
- Editar o eliminar oferta: Editar la información de la oferta de prácticas o eliminarla.
- Acceder a las ofertas publicadas: Acceder al listado de ofertas publicadas por la empresa.
- Gestionar candidaturas: Acceder al listado de las candidaturas recibidas, consultar el detalle de estas y cambiar el status de esta.

#### **Casos de uso para Usuario no registrado:**

- Buscar y filtrar ofertas: Acceder a las ofertas publicadas y filtrarlas según varios parámetros.
- Ver detalle de las ofertas: Acceder a la información extendida de cada oferta.



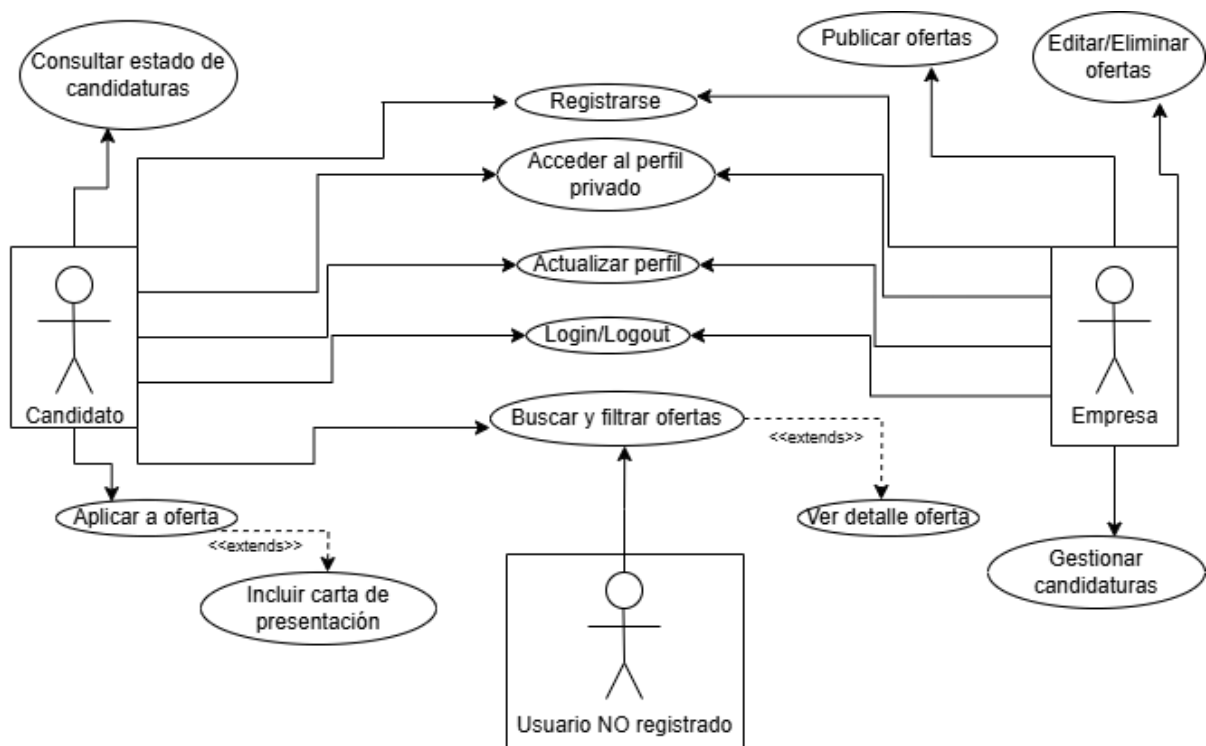


Figura 6 - Diagrama de casos de uso.

## 7. Diseño del proyecto

### Diseño de estilo:

El diseño del estilo InternJobs, a diferencia del diseño de la base de datos y arquitectura de la aplicación, se ha desarrollado de forma práctica directamente durante la implementación del frontend, sin utilizar herramientas de diseño gráfico como Figma o Adobe XD. Esta decisión respondió a un enfoque ágil y funcional, centrado en la eficiencia y en la claridad visual desde el inicio.

El hecho de contar con cierta experiencia con la utilización TailwindCSS, así como la facilidad de uso aportada por este framework de estilos fué determinante en la decisión de crear el estilo durante la implementación.

El uso de TailwindCSS facilitó la creación, de manera rápida y natural, de una interfaz moderna, limpia y responsiva. En lugar de definir maquetas estáticas previas, se optó por construir el diseño “sobre la marcha”, permitiendo realizar ajustes visuales inmediatos a medida que se implementaban los distintos componentes de la aplicación.

Principales decisiones del diseño:

- **Modo oscuro como base:** Se utilizó un fondo de base bg-gray-800, lo que proporciona un estilo oscuro y moderno. Esta decisión aporta un contraste visual que resalta los elementos interactivos y mejora la legibilidad en entornos de poca luz.
- **Estilo minimalista y profesional:** Se buscó construir una interfaz clara y funcional, sin demasiados elementos decorativos innecesarios. El foco está en la usabilidad y en la jerarquía visual de los contenidos.
- **Responsive design:** Gracias a la utilidades de tailwind, se logró que los elementos fueran adaptables a distintos tamaños de pantalla, optimizando la experiencia de usuario tanto en escritorio como en dispositivos móviles.
- **Componentes reutilizables:** Se desarrollaron componentes visuales reutilizables como formularios, tarjetas de ofertas, botones y banners, con estilos coherentes y definidos mediante clases de utilidad.
- **Tarjetas y bloques de estilo semi-redondeado:** Muchos de los elementos visuales como formularios, tarjetas, contenedores y botones, presentan bordes semi-redondeados (rounded o rounded-lg), aportando estética moderna y suave de estilo similar a Windows 11. Este diseño contribuye a un diseño profesional pero algo desenfadado.
- **Jerarquía visual clara y alto contraste:** Para garantizar la jerarquía visual efectiva con buena accesibilidad, se emplearon combinaciones de color específicas. Los textos principales utilizan text-gray-100 sobre fondos oscuros (bg-gray-800 y bg-gray-900), creando un contraste elevado y agradable a la vista. Además se incorporaron colores como bg-emerald en las intensidades 400, 500 y 600 para botones, enlaces o elementos destacados, asegurando que las llamadas a la acción sean fácilmente identificables y visualmente agradables.

Este enfoque directo, apoyado completamente en TailwindCSS, permitió un desarrollo fluido, flexible e iterativo, ajustando la estética y funcionalidad de la interfaz según las necesidades del proyecto en cada etapa.

### Arquitectura del sistema:

**InternJobs** sigue una arquitectura MVC (Modelo-Vista-Controlador), que facilita la organización del código, la separación de responsabilidades y el mantenimiento del proyecto a medio y largo plazo.

Esta Arquitectura ha aplicado de manera clara separando los bloques:

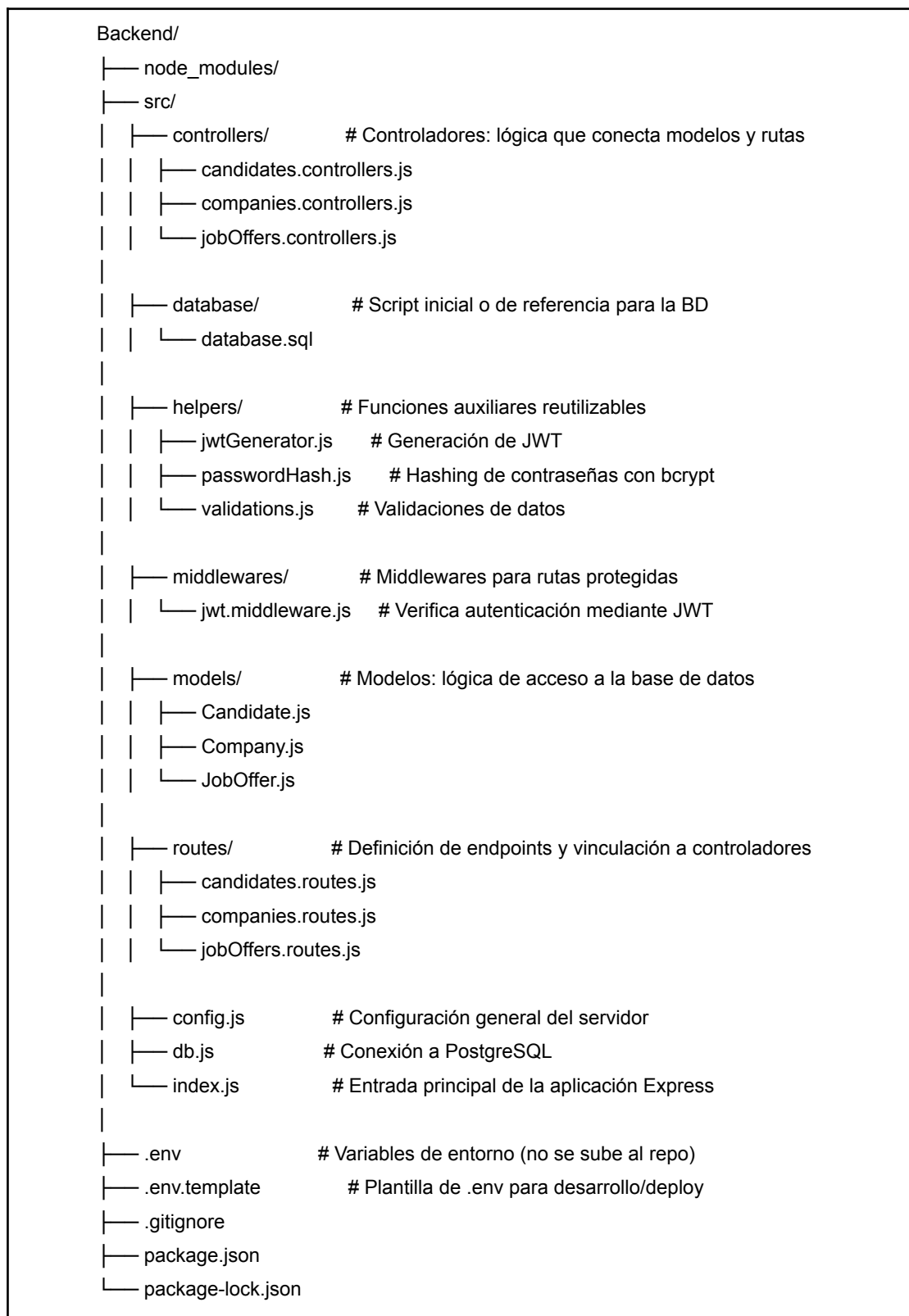
- **Backend (Modelos y Controladores):** Responsable de la parte de la lógica de negocio, acceso a la base de datos y seguridad.
- **Frontend (Vistas):** Implementando una aplicación de página única (SPA) desarrollada en Vue3, que consume la API del backend para construir directamente la interfaz de usuario.

### **Backend:**

El backend ha sido desarrollado con [Node.js](#) y el framework **Express**, utilizando una estructura modular, basada en controladores, modelos y rutas. A continuación se detallan los aspectos clave:

- **Consultas SQL en los modelos:** La comunicación de datos hacia PostgreSQL se realiza mediante consultas directas (raw queries), sin usar ORM. Estas consultas se estructuran utilizando el sistema seguro de PostgreSQL basado en parámetros: Se envía la consulta con marcadores de posición (\$1, \$2, etc.) y un array con los valores, lo que protege frente a inyecciones SQL.
- **Controladores:** Los controladores reciben y procesan las peticiones, llamando a los modelos correspondientes para ejecutar consultas SQL. Posteriormente, formatean y devuelven la respuesta adecuada al cliente.
- **Seguridad y autenticación:** Las contraseñas se almacenan de forma segura gracias al uso de bcrypt que se encarga del hash y los saltos de las contraseñas. Se utilizan JSON Web Tokens para autenticar a los usuarios. Las rutas protegidas están gestionadas por middlewares personalizados, que validan la autenticidad del token antes de permitir acceso a controladores.
- **API REST:** Todo el backend está estructurado como una API RESTful, lo que permite una integración limpia y clara con el frontend.

Diagrama de estructura del backend:

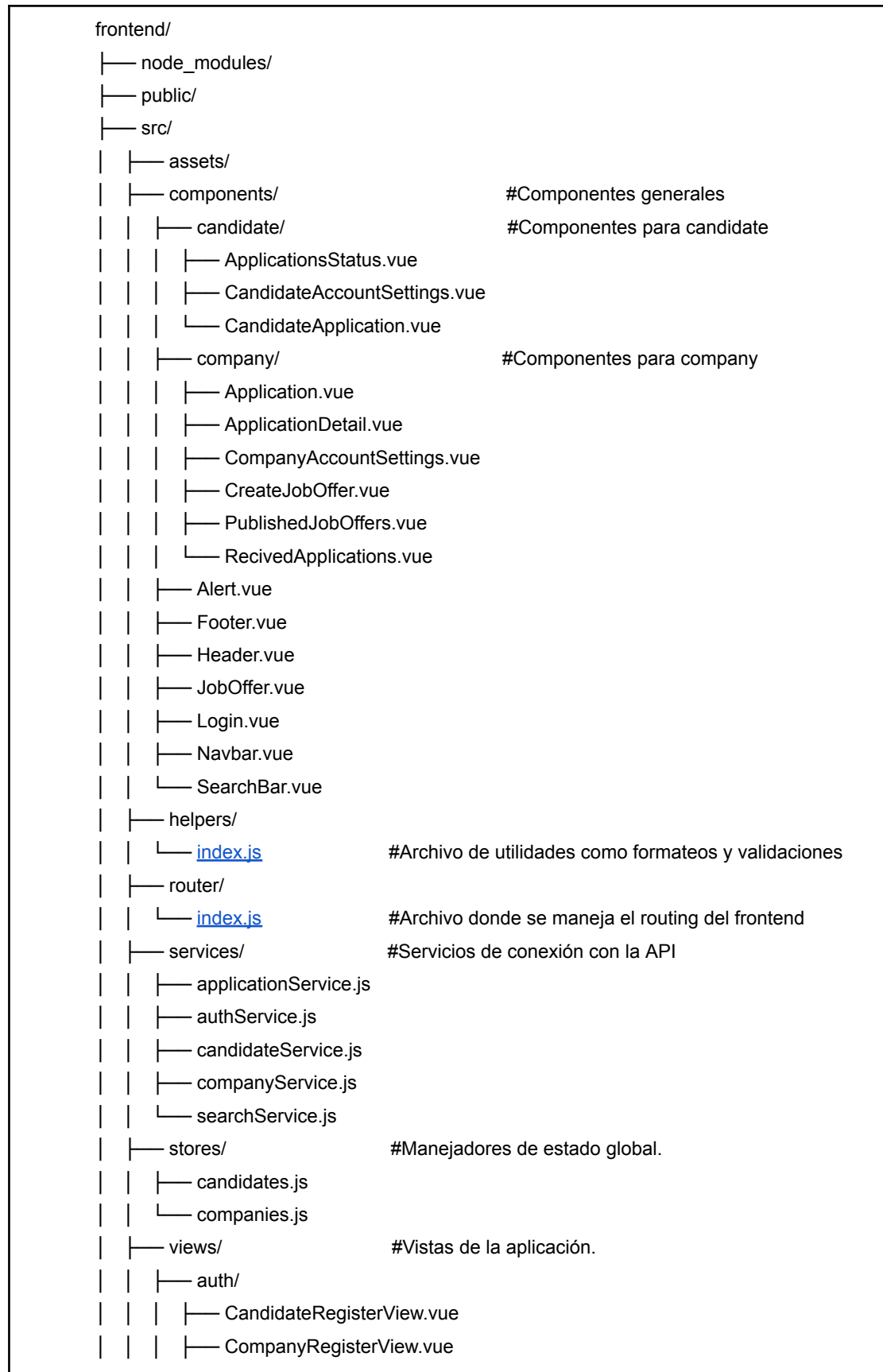


**Frontend:**

El frontend está construido como una Single Page Application (SPA) usando Vue 3, lo que permite una experiencia de usuario fluida y rápida sin recargas completas de la página.

- **Navegación y routing:** Se utiliza Vue Router para gestionar la navegación entre vistas sin salir de la aplicación. Las rutas están definidas con nombres específicos y permiten el paso de parámetros, lo que facilita una navegación clara y mantenible, ya que, evita el uso directo de strings en los “paths” y permite refactorizar rutas de forma segura.
- **Manejo del estado:** El estado global de la aplicación se realiza con Pinia, lo que facilita el control de estados, como la sesión de usuario, los datos son cargados desde la API y otros elementos compartidos entre componentes. Esto ha facilitado la adaptación de la barra de navegación al usuario logueado en todas las vistas con persistencia del usuario incluso con recarga de página.
- **Comunicación con el backend:** El frontend se comunica con el backend mediante peticiones HTTP a la API REST. Para estructurar esta interacción de manera clara y reutilizable se han creado servicios adaptados usando Fetch API. Estos servicios encapsulan las llamadas a los distintos endpoints del backend, proporcionando una capa de abstracción que permite centralizar la lógica de comunicación y mejorar la mantenibilidad del código.

## Diagrama de estructura del Frontend:



```
| | | |— LoginCandidateView.vue
| | | |— LoginCompanyView.vue
| | | |— profile/
| | | |— ApplyFormView.vue
| | | |— HomeView.vue
| | | |— JobOfferView.vue
| | | |— SearchResultsView.vue
| | |— App.vue
| |— main.js
|— .env.local
|— .env.template
|— .gitignore
|— index.html
|— jsconfig.json
|— package-lock.json
|— package.json
|— vite.config.js
|— README.md
```

## 8. Despliegue y pruebas

### Despliegue:

Existen varias plataformas donde podemos hacer el despliegue a producción de **InternJobs**, los ejemplos más populares para desplegar el frontend son **Netlify** o **Vercel**. Para desplegar el backend y la base de datos opciones muy populares son **Railway**, **Render** o **SupaBase**. Todas las plataformas mencionadas ofrecen una capa gratuita que incluye el posible despliegue de esta aplicación, aunque, el rendimiento del alojamiento suele ser bajo en este tipo de suscripción.

Debido a que el despliegue a una de estas plataformas conlleva más pasos y pueden variar entre las distintas plataformas, se puede consultar la sección [12.2. Manual de instalación](#) donde detallo los pasos para hacer el despliegue de forma local.

### Pruebas:

Durante el desarrollo de InternJobs se utilizó Postman para realizar pruebas exhaustivas de todos los endpoints definidos en la API REST del backend. Estas pruebas permitieron validar el correcto funcionamiento de las rutas, los controladores y la lógica de acceso y manipulación de datos. Se agrupan por entidad:

COMPANIES

- **POST /loginCompanie**  
Prueba de autenticación con credenciales válidas y erróneas. Verificación del token JWT recibido.
- **POST /createCompanie**  
Registro de empresas con datos correctos e intentos con campos incompletos o duplicados.
- **GET /getCompanie**  
Consulta del perfil de empresa autenticada mediante token.
- **PUT /updateCompany**  
Prueba de actualización de datos con validación de cambios persistentes en la base de datos.
- **DELETE /deleteCompanie**  
Eliminación de una empresa autenticada. Confirmación de borrado y protección de rutas.
- **POST /createJobOffer**  
Creación de ofertas asociadas a una empresa autenticada. Verificación de inserción correcta.
- **GET /getJobOffersByCompany**  
Recuperación de ofertas publicadas por una empresa específica.
- **GET /getJobOfferByIdCom**  
Consulta de una oferta concreta por ID, controlando acceso desde una empresa válida.
- **PUT /updateJobOffer**  
Modificación de una oferta ya publicada.
- **DELETE /deleteJobOffer**  
Eliminación de una oferta publicada por la empresa correspondiente.
- **GET /getApplicationsByCompany**  
Listado de candidaturas recibidas para las ofertas de una empresa.
- **GET /getApplicationById**  
Consulta individual de una candidatura específica.
- **PATCH /updateApplication**  
Actualización del estado de una candidatura (por ejemplo, "aceptada" o "rechazada").



## CANDIDATES

- **POST /loginCandidate**  
Login con credenciales válidas e inválidas. Validación de JWT recibido.
- **GET /getCandidate**  
Consulta de datos del candidato autenticado.
- **POST /createCandidate**  
Registro de nuevos candidatos. Verificación de duplicados y validación de campos.
- **DELETE /deleteCandidate**  
Eliminación de cuenta de candidato.
- **PUT /updateCandidate**  
Actualización de datos personales y académicos.
- **GET /getApplicationsByCandidate**  
Listado de todas las candidaturas realizadas por el candidato autenticado.

## JOB OFFERS

- **GET /getJobOffers**  
Recuperación de todas las ofertas activas.
- **GET /getJobOfferById**  
Consulta del detalle de una oferta pública.
- **POST /applyToJobOffer**  
Aplicación a una oferta por parte de un candidato autenticado.
- **GET /getJobOffersByParams**  
Filtro y búsqueda de ofertas según parámetros como título, localidad o modalidad.

Una vez superadas las pruebas sobre los endpoints de la API y desarrollado el frontend, se realizaron las pruebas funcionales completas del flujo de uso desde la perspectiva de los dos tipos de usuario: candidato y empresa.

## CANDIDATO

- Registro de nuevo candidato.
- Login con credenciales válidas.
- Visualización y edición del perfil.
- Navegación y filtrado de ofertas.
- Visualización del detalle de una oferta.
- Aplicación a una oferta.
- Revisión del historial de candidaturas enviadas.
- Logout y validación de seguridad de rutas privadas.

## EMPRESA

- Registro de nueva empresa.
- Login con credenciales válidas.
- Edición del perfil de empresa.
- Creación de nuevas ofertas de prácticas.
- Visualización, edición y eliminación de ofertas ya publicadas.
- Consulta de candidaturas recibidas por cada oferta.
- Evaluación de candidaturas mediante actualización de estado.
- Logout y verificación del acceso restringido a secciones privadas.

Todas las pruebas fueron fundamentales para detectar y corregir errores, asegurar la lógica de negocio, verificar la protección de rutas privadas con JWT, y garantizar una experiencia fluida tanto para empresas como candidatos.

## 9. Conclusiones

El desarrollo de **InternJobs** ha supuesto un gran desafío para mí, ya que debido a mi situación personal no he contado con demasiado tiempo para el desarrollo de esta, y sin embargo, hago uso de tecnologías en las que apenas tengo experiencia, un ejemplo de ello es Vue 3. No obstante, ya contaba con un amplio recorrido con JavaScript y alguna que otra aplicación creada con [Node.js](#).

Después de un análisis exhaustivo sobre el tipo de aplicación que quería crear y que funcionalidad iba a tener, me decidí de manera relativamente rápida por [Node.js](#) y Express en la parte del backend, ya que eran tecnologías que había utilizado con anterioridad y se adaptaban muy bien al tipo de aplicación que quería desarrollar, pero por otro lado, en la parte del frontend nunca tuve una idea clara de qué tecnología se adaptaba mejor a la idea que quería construir, aunque siempre tuve claro que necesitaría un framework o librería de JavaScript para llevar a cabo mi idea, debido a que necesitaba routing y gestión del estado global. Dado que no tenía mucha experiencia con frameworks frontend de JavaScript, decidí investigar un poco sobre los más populares, finalmente me decidí por [Vue.js](#) ya que contaba con todo lo que necesitaba mi aplicación y ofrecía una curva de aprendizaje “asumible”, aunque una vez había empezado el frontend me di cuenta que “asumible” en programación es muy relativo.

Con todo, puedo decir que la creación de **InternJobs** ha supuesto un viaje de aprendizaje continuo, tanto para afianzar tecnologías ya descubiertas con anterioridad como para crear mi primer frontend en Vue 3. Enfrentarme al desarrollo de mi primera aplicación real full-stack con separación de responsabilidades ha sido un desafío completamente enriquecedor, que ha marcado una gran diferencia con las aplicaciones que había creado con anterioridad.

La aplicación ha sido desarrollada siguiendo buenas prácticas, con código limpio, mantenible y escalable. Considero que sigue con buena parte de los principios SOLID, y clean code, aunque, adaptados a la tipología de la aplicación y a la amplitud de mi conocimiento.

Durante el desarrollo de la aplicación me he enfrentado a diversas dificultades. En la parte del backend primeramente quise implementar un sistema sin bcrypt ni JWT para mantener la sencillez en el desarrollo y obtener un MVP lo antes posible, a pesar de que quería añadir esa funcionalidad más adelante. Rápidamente pude comprobar que tratar de evitar esa funcionalidad, en realidad, estaba aumentando la complejidad más que no haberlas incluido desde el principio, ya que acababa teniendo urls muy complejas, arquitectura más compleja,

imposibilidad de seguir con los principios REST y menos seguridad en la aplicación. Pude descubrir por qué la sencillez y utilidad de JWT, hacen a esta tecnología la solución predilecta para aplicaciones con separación de responsabilidades.

Para terminar con las conclusiones de backend quiero comentar que a pesar del inconveniente anterior, pude notar por qué Express es una tecnología tan utilizada en APIs REST de aplicaciones comerciales, el desarrollo de la API me resultó muy natural y fácil de entender, y pude ir desarrollando endpoints de manera completamente programada.

El primer contratiempo que tuve en el desarrollo del frontend fué la sincronización de la barra de navegación con cada usuario y tipo de usuario de manera dinámica y persistente. Para mí la barra de navegación era el eje central de mi aplicación, ya que iba a ser un componente compartido por todas las vistas, y quería que tuviera un comportamiento reactivo al login y logout y al rol del usuario que hacía login. Puesto que para estas alturas del desarrollo ya no quería hacer cambios en la base de datos, y no había incorporado un campo "role" a las tablas de "candidates" y "companies", fué necesario forzar el guardado de este dato en localStorage dentro de la funcionalidad del login. Aunque aún tuve más problemas de sincronización, ya que tuve que modificar el endpoint de login, tanto de candidates como de companies, debido a que la sincronización de la navbar fallaba al estar consumiendo la id de sus respectivas tablas y el endpoint no estaba devolviendo este dato. Estoy satisfecho con el resultado del desarrollo, no tanto por la aplicación creada, si no por todo el proceso de aprendizaje que ha supuesto, ha sido un proceso enriquecedor que creo que no habría sido posible sin la necesidad de presentar un proyecto final.

El desarrollo de **InternJobs** no termina aquí, a pesar de tener la versión entregable ya lista, el desarrollo va a continuar con todos los aspectos que quiero mejorar y todas las funcionalidades que pretendo añadir, todos los detalles de estos aspectos están detallados en el siguiente apartado.

## 10. Vías futuras

En este apartado comentaré todos los aspectos a mejorar y las nuevas funcionalidades que añadiré a la aplicación después de su entrega, y que por falta de tiempo aún no han podido ser implementadas.

A pesar de que la aplicación en el momento de la entrega ya cumple con todos los objetivos, no dejo de notar que aún podría añadir estas funcionalidades y pulir ciertos aspectos, de manera que las mejoras planteadas son las siguientes:

- **Confirmación de password en las funciones de actualizar datos:** Para este punto los usuarios pueden actualizar su información privada, así como las empresas pueden gestionar candidaturas, ofertas. Todas estas funcionalidades se hacen desde el panel privado de cada usuario logueado en rutas protegidas, pero es buena práctica a nivel de seguridad añadir una confirmación de password en cada actualización o creación de nuevos datos.
- **Validaciones adicionales en el frontend:** En este punto se cuenta con validaciones extensas a nivel de backend, y validaciones más suaves en el frontend ya que se confía en la primera capa de validación propia de HTML. De cara a próximas mejoras se prevé añadir validación más estricta en el frontend, y tanto en frontend como en backend, una validación más estricta de password adaptada a los estándares modernos, de mínimo 12 caracteres, mayúsculas y minúsculas, caracteres y números.
- **Estilo responsive en dashboard privado:** A pesar de que en este punto la mayoría de la aplicación ya cuenta con un diseño responsive, la parte de perfil privado se ha diseñado con estilo dashboard de escritorio, con barra lateral izquierda y contenido principal a la derecha. Preveo rediseñar esta parte con estilos de dashboard mobile también, con contenido vertical.
- **Paginación en resultados de búsqueda:** Probablemente la funcionalidad más desafiante sea la paginación de los resultados de búsqueda, ya que requiere ajustes tanto en el backend como en el frontend. Dado que las pruebas en local no han llenado la base de datos lo suficiente como para requerir paginación, es una funcionalidad que debe estar en la aplicación dado que no tenerla es una limitación a

la escalabilidad de esta. También sería interesante incluir paginación en la gestión de candidaturas y listado de ofertas publicadas en el perfil de empresa.

- **Sistema de confirmación de cuentas por email:** Es bastante habitual en las aplicaciones modernas encontrarnos un sistema de confirmación de cuenta a través de tokens que se envían al email del futuro usuario de esa cuenta, este sistema es una buena práctica ya que evita que nuestra aplicación se llene de “bots” con cuenta de usuario ocupando recursos de nuestra aplicación. Esta es una funcionalidad que preveo añadir en el caso de desplegar la aplicación a producción, pero en el caso del desarrollo local puede dificultar la evaluación ya que puede alargar en exceso las pruebas de registro de usuarios.
- **CV en formato PDF:** Aunque en este momento la aplicación ya permite guardar el CV de los candidatos, este documento se guarda como texto plano, lo ideal sería permitir a los candidatos subir su CV en formato pdf. La decisión de no incluir esta funcionalidad era poder tener un MVP entregable lo antes posible debido al tiempo limitado, y esta funcionalidad habría conllevado un tiempo dilatado ya que guardar este tipo de documentos en la base de datos no es lo más óptimo en términos de rendimiento y habría que haber buscado un alojamiento cloud para guardar este tipo de documentos y trazar una arquitectura vía URLs. Lo que a su vez supondría una posible violación de la normativa de entrega, ya que habría necesitado de servicios mencionados en la normativa como no usables o de uso limitado y aprobado previamente por el profesor.
- **Imágenes de perfil:** Finalmente permitir a los usuarios incluir imágenes de perfil habría sido una funcionalidad ideal, y creo que cerraría perfectamente la aplicación que he construido ya que permitiría mostrar su imagen a los candidatos, y a las empresas mostrarse con su imagen corporativa que podría aparecer en los resultados de búsqueda y ser fácilmente identificable por los candidatos. Los motivos por los que aún no se ha incluido esta funcionalidad son idénticos a los del anterior punto.

## 11. Bibliografía y Webgrafía

A continuación detallaré todos los recursos que de alguna forma han nutrido o han ayudado a dar forma al proyecto final.

- Juan Pablo de la Torre Valdez. (2025). [Vue.js](#) 3 - La guía completa - Composition Pinia MEVN 10 Apps.  
Fuente: <https://www.udemy.com/course/vuejs-la-guia-completa-composition-pinia-mevn-creando-proyectos-reales>
- Juan Pablo de la Torre Valdez. (2025). - JavaScript Moderno Guía Definitiva.  
Fuente: <https://www.udemy.com/course/javascript-moderno-guia-definitiva-construye-10-proyectos>
- Fernando Herrera. (2025). - [Node.js](#) de Cero a Experto.  
Fuente: <https://cursos.devtales.com/courses/nodejs-de-cero-a-experto>
- Bluuweb. (2025). - Tutorial de [Node.js](#) + [Express.js](#) + JWT + POSTGRES.  
Fuente: <https://www.youtube.com/watch?v=5gLZ0Xzzmds&t=4535s>
- GitHub, Inc. (2025). GitHub. Fuente: <https://github.com/>
- bcryptjs. (2025). Fuente: <https://www.npmjs.com/package/bcryptjs>
- jsonwebtoken. (2025). Fuente: <https://www.npmjs.com/package/jsonwebtoken>
- OpenJS Foundation. (2025). Node.js. - Ejecuta JavaScript en cualquier parte.  
Fuente: <https://nodejs.org/es>
- The PostgreSQL Global Development Group. (2025). Fuente: <https://www.postgresql.org/>
- Postman. (2025). Postman: The World's Leading API Platform | Sign Up for Free.  
Fuente: <https://www.postman.com/>

- TablePlus. (2025). Fuente: <https://docs.tableplus.com/>
- Notion Labs Inc. (2025). Notion - Fuente: <https://www.notion.so/>

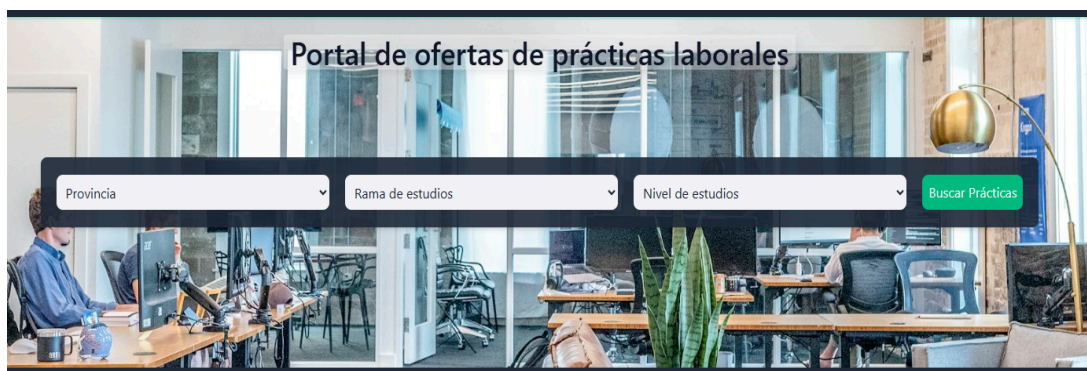
## 12. Anexos

### 12.1. Manual de usuario

Como ya he comentado anteriormente existen dos tipos de usuarios registrados, los candidatos y las empresas, y los usuarios no registrados. Todos pueden acceder a la funcionalidad de búsquedas y obtención de ofertas y visualización del detalle de las ofertas. El resto de funcionalidades son específicas para cada tipo de usuario registrado.

#### **Búsqueda, filtrado y detalle de ofertas:**

Posicionarse en la barra de búsqueda y elegir los parámetros por los que se quiere filtrar, o en caso de no querer filtrar no seleccionar ninguno y clicar el botón de “Buscar Prácticas”. Se obtendrán una lista de resultados separados por tarjetas, para acceder al detalle de cada oferta clicar sobre la tarjeta de esa oferta.





<b>Técnico de rayos X.</b> Udemy Online Se requiere estudiante de medicina nuclear para hace radiografías. <b>SANIDAD</b> <b>CFGS</b> Incentivo: 200€	31/05/2025 <b>Sevilla</b>
<b>Desarrollador COBOL</b> Udemy Online Estamos buscando estudiante con perfil IT para formarse en COBOL. <b>IT</b> <b>CFGM</b> Incentivo: 200€	31/05/2025 <b>Barcelona</b>
<b>Profesor de Python</b> Udemy Online Se requiere un profesor de python y creador de contenido online, con 5 años de experiencia. <b>IT</b> <b>CFGS</b> Incentivo: 600€	28/05/2025 <b>Barcelona</b>

**Técnico de rayos X.**  
Udemy Online  

Sevilla

SANIDAD

200€

CFGS

Publicada: 31/05/2025

**Descripción:**  
Se requiere estudiante de medicina nuclear para hace radiografías.  
[Haz login como candidato para aplicar a esta oferta](#)

### Registro como Candidato/Empresa:

A través de los botones situados a la derecha de la barra de navegación pulsar el botón de “Regístrate”. Dependiendo del rol con el que el usuario quiera registrarse navega entre los formularios de candidato y empresa situados en la parte superior derecha del formulario con el texto de “Soy una empresa” y “Soy un candidato”. Seguir los pasos de formulario y finalmente pulsar el botón de “Crear Cuenta”

Regístrate como candidato en InternJobs

Nombre:

Email:

Teléfono:

Currículum en texto:

4000

Password:

Confirma Password:

[Crear Cuenta](#)

### Login:

Acceder a los botones de la parte derecha de la barra de navegación con el texto “Acceso Empresas” o “Acceso Candidatos” dependiendo del rol con que el usuario haya sido registrado, completar el breve formulario de email y password y pulsar el botón de “Entrar a mi cuenta”.

Acceso candidatos

Email:

Password:

[Entrar a mi cuenta](#)

**Logout:**

Una vez realizado el Login aparecerán dos botones en la parte derecha de la barra de navegación, para realizar el logout simplemente pulsar el botón rojo con texto “Logout”.

**Acceso al perfil privado:**

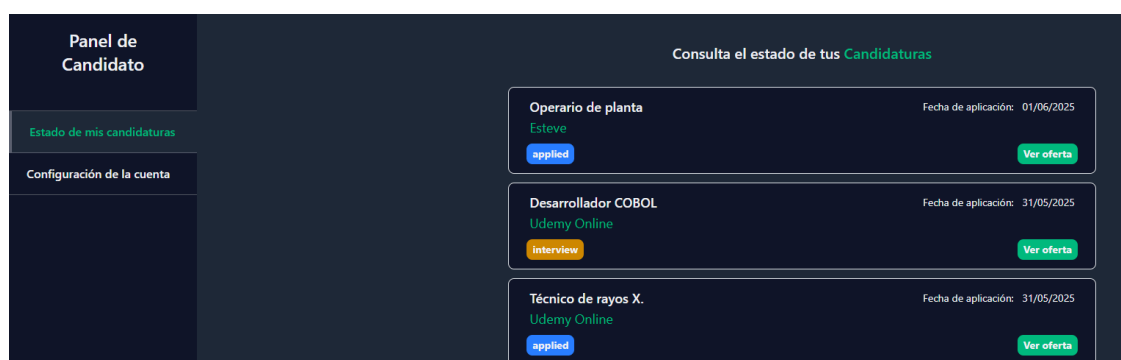
Pulsar el botón de la parte derecha de la barra de navegación con el nombre del usuario registrado y un icono a la izquierda de este.

**Actualización de datos de usuario:**

Acceder a la pestaña de la barra lateral izquierda donde indica “Configuración de la cuenta” realizar los cambios deseados dentro del formulario de información y pulsar el botón “Guardar Cambios”.

**Estado de las candidaturas (Candidatos):**

Acceder a la pestaña que indica “Estado de mis candidaturas” en el menú lateral izquierdo, se ofrece una lista de las ofertas a las que el usuario ha aplicado y el estado del proceso de selección en el que se encuentran. Para acceder al detalle de la oferta pulsar en el botón “Ver Oferta”.

**Listado de ofertas publicadas (Empresas):**

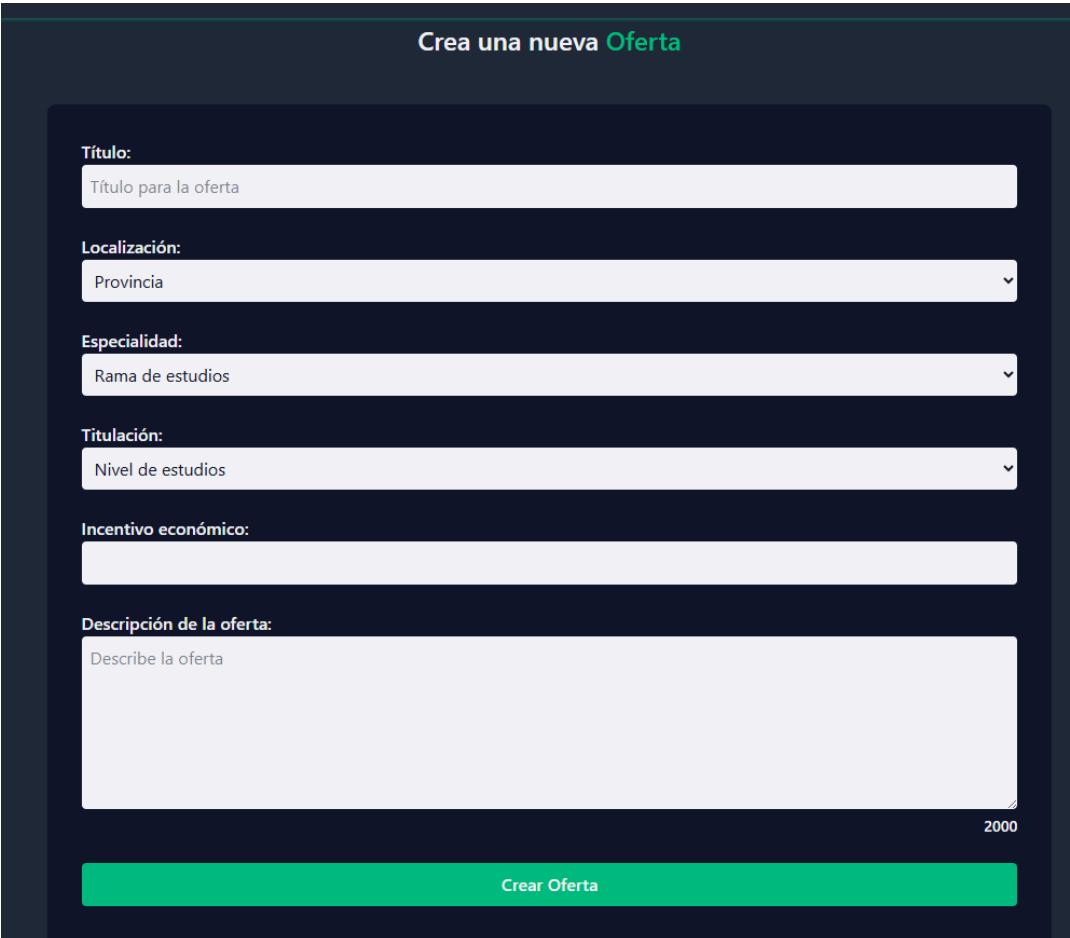
Acceder a la pestaña de “ofertas publicadas” en el menú lateral izquierdo. Se ofrece el listado de ofertas publicadas.

**Actualizar o eliminar oferta publicada:**

Pulsar el botón de “Actualizar/Eliminar oferta”, la aplicación te redirecciona al formulario de la oferta, aplicar los cambios deseados y pulsar el botón “Guardar Cambios”.

**Crear oferta (Empresas):**

Acceder en el menú lateral izquierdo a la pestaña de “Crear oferta”, será visible el formulario con los datos necesarios para crear la oferta. Rellenar los datos y pulsar el botón “Crear Oferta”.



Formulario para crear una nueva oferta. El formulario está titulado "Crea una nueva Oferta" y contiene los siguientes campos:

- Título:** Campo de texto con el placeholder "Título para la oferta".
- Localización:** Selector de provincia con "Provincia" como opción visible.
- Especialidad:** Selector de rama de estudios con "Rama de estudios" como opción visible.
- Titulación:** Selector de nivel de estudios con "Nivel de estudios" como opción visible.
- Incentivo económico:** Campo de texto vacío.
- Descripción de la oferta:** Área de texto grande con el placeholder "Describe la oferta".

En la parte inferior del formulario hay un botón verde que dice "Crear Oferta".

**Gestión de candidaturas (Empresas):**

Pulsar en la pestaña de “Candidaturas recibidas”, se ofrecerá la lista de candidaturas recibidas a ofertas publicadas por el usuario, así como un resumen de la información del candidato y el estado del proceso de selección con un sistema de colores. Para actualizar el estado pulsar en “Gestionar aplicación” donde se puede ver el detalle de la candidatura, carta de presentación y CV del candidato. En la parte de abajo

donde indica “Actualizar el estado de la aplicación” seleccionar el estado al que se quiere actualizar la candidatura y pulsar en “Actualizar”.

### Gestiona las Candidaturas recibidas

Nombre: Candidato 1

Aplicó a: Desarrollador COBOL

Email: oscar.serrano91@gmail.com

interview

Aplicó el: 31/05/2025

Gestionar aplicación

Nombre: Candidato 1

Aplicó a: Técnico de rayos X.

Email: oscar.serrano91@gmail.com

applied

Aplicó el: 31/05/2025

Gestionar aplicación

Nombre: Candidato 1

Aplicó a: Profesor de Python

Email: oscar.serrano91@gmail.com

rejected

Aplicó el: 28/05/2025

Gestionar aplicación

### Desarrollador COBOL

Candidato 1

987654321

oscar.serrano91@gmail.com

interview

#### Carta de presentación

Soy un entusiasta de COBOL, y tengo experiencia en empresas del sector financiero.

#### CV

Full stack developer PERFIL Desarrollador de aplicaciones web en formación, con diez años de experiencia en industria farmacéutica, con el objetivo de migrar hacia el sector IT. Tras una década dedicada al análisis riguroso en entornos farmacéuticos, ahora estoy canalizando mi pasión por la tecnología hacia el desarrollo de aplicaciones web. Mi formación actual me está proporcionando las herramientas necesarias para construir soluciones tecnológicas eficientes, seguras y escalables. Mi objetivo actual es poder combinar el rigor científico, adquirido durante estos años, con el dinamismo y creatividad de la industria tecnológica. CONTACTO Teléfono: 637110121 Email: oscar.serrano91@gmail.com Localidad: Viladecans (Barcelona, España) Portafolio: www.oserranom.pro SKILLS JavaScript Python SQL HTML CSS Node.js Git/GitHub IDIOMAS Castellano (Nativo) Catalán (Nativo) Inglés (B2) Formación académica FORMACIÓN ACADÉMICA 2024: CFGS Desarrollo de aplicaciones web (Centro de formación Ilerna) 2020: Grado en Química (Universidad Nacional de Educación a Distancia) EXPERIENCIA LABORAL 2015 - Actualidad: Synthon Hispania - QC Analyst Análisis de muestras, revisión de datos analíticos, cálculo de tendencias y estadísticas del comportamiento de los productos farmacéuticos, registro de datos analíticos en software LIMS, actualidad y gestión de stock. Cursos de formación CURSOS DE FORMACIÓN 2024: Píldoras Informáticas - JavaScript Tutorizado 120 h

#### Actualiza el estado de la aplicación

Applied

Actualizar

## 12.2. Manual de instalación

### Paso 1: Instalación de [Node.js](#) y PostgreSQL

Para poder levantar la aplicación en modo local necesitaremos tener instalados en nuestra máquina [Node.js](#) v22 o superior y PostgreSQL v16 o superior. Si aún no disponemos de estas instalaciones se pueden descargar a través de los enlaces <https://nodejs.org/es/download> para [Node.js](#), y <https://www.postgresql.org/download> para PostgreSQL.

Una vez instalados podemos comprobar que las versiones instaladas son las versiones indicadas o superiores con los comandos:

```
PS C:\Users\Oscar> node -v
v22.14.0
PS C:\Users\Oscar> psql --version
psql (PostgreSQL) 17.3
PS C:\Users\Oscar>
```

### Paso 2: Configurar las variables de entorno:

#### Backend:

Dentro del fichero de backend navega hasta el documento “.env.template” y sustituye las variables de entorno por las de tu conexión local a PostgreSQL. A continuación sustituye el nombre del documento por simplemente “.env”.

#### Frontend:

Dentro del fichero de frontend navega hasta el documento “.env.template” y configura la variable de entorno con la ruta del puerto donde esté configurada la conexión local del backend, por ejemplo “<http://localhost:4000/api>”, a continuación sustituye el nombre del documento por “.env.local”.

### **Paso 3: Instalación dependencias**

A través de la terminal navegar hasta el fichero “Backend” del proyecto y ejecutar el comando “npm i”.

Volver a la raíz del proyecto y navegar hasta el fichero “Frontend” y ejecutar el comando “npm i”.

### **Paso 4: Levantar el servidor del Backend y el Frontend del proyecto**

A través de la terminal acceder al fichero de Backend y ejecutar el comando “npm run dev”. A continuación volver a la raíz del proyecto y acceder al fichero de Frontend y ejecutar de nuevo el comando “npm run dev”. Acceder a la url ofrecida por Vite, por defecto “<http://localhost:5173/>”.